Vol.2013-HPC-141 No.4
2013/9/30

京、FX10 及び CX400 におけるブラソフコードの性能チューニング

梅田 隆行[†1]　　深沢 圭一郎[†2]

ブラソフコードは宇宙空間を満たす無衝突プラズマの第一原理シミュレーション手法である．ブラソフシミュレーションでは，位置及び速度で与えられる超多次元位相空間における荷電粒子の分布関数の時間発展を，運動論方程式により直接解き進めている．4 次元以上の空間を扱うシミュレーションでは，ノードあたり，あるいはコアあたりに使用できるメモリ容量の制限から，数値解法や性能チューニングにおいて様々な工夫が必要である．本研究グループはこれまでに様々な HPC 関連プロジェクトと通じて，ブラソフコードの性能チューニングを行ってきた．本論文では特に，SPARC プロセッサ（京コンピュータ，Fujitsu FX10，FX1）及び Xeon プロセッサ（Fujitsu CX400）におけるブラソフコードの性能評価により培ってきた，アーキテクチャ特有及び共通のチューニング手法について議論を行う．

Performance tuning of Vlasov code for space plasma simulations on K, FX10 and CX400

TAKAYUKI UMEDA[†1]　KEIICHIRO FUKAZAWA[†2]

Vlasov code is a first-principle simulation method for collisionless space plasma. The Vlasov code solves the time development of phase-space distribution functions of charged particles in hyper-dimensions based on fully kinetic equations. Since the distribution functions are defined in more than four dimensions, the Vlasov code requires high-resolution and high-performance numerical schemes which should work in limited computational memory per node or per core. In this paper, performance tuning of massively-parallel Vlasov code on various scalar CPU architectures, such as the SPARC processors (K-computer, Fujitsu FX10, Fujitsu FX1) and the Xeon processors (Fujitsu CX400) is discussed based on the results of our performance measurement studies conducted under Japanese HPC projects.

1. はじめに

我々が住む宇宙の 99.99%以上の体積はプラズマと呼ばれる電離気体で占められている．宇宙空間に存在するプラズマの大部分は密度が非常に小さく無衝突状態にあり，宇宙プラズマ（無衝突プラズマ）を理解することは，宇宙の本質的な理解につながる．

我々が住む地球周辺の宇宙環境は，太陽から放出された高速のプラズマ流である太陽風及び太陽風が運ぶ惑星間空間磁場（太陽の固有磁場）と，地球の固有磁場との相互作用によって複雑な磁気圏構造を形成している．プラズマ放出現象をはじめとする太陽の様々な変動により，宇宙飛行士の被曝，人工衛星の故障や通信障害に繋がる地球磁気圏・電離圏の環境変動が引き起こされ，これを宇宙天気と呼ぶ．近年の国際宇宙ステーションでの活動や人工衛星の打ち上げなど，日本においても宇宙利用が現実的になってきており，宇宙天気の予報・予測に繋がる宇宙プラズマ研究は極めて重要である．

地球磁気圏内には，プラズマの密度や温度などの物理パラメータが異なる様々な領域が生じる．その領域間の境界層で現れる不安定性（平衡状態の破れ）は，磁気圏の変動に大きな影響を与えていると考えられている．グローバル磁気圏構造に対して，境界層不安定性は中間（メゾ）スケール現象と呼ばれる．これらのグローバル及び中間スケールの現象は，粒子運動論を扱う方程式であるブラソフ（無衝突ボルツマン）方程式の 0 次・1 次・2 次のモーメントを取ることによって求められる磁気流体力学（MHD）方程式によって記述される．しかし，近年の科学衛星による高精度な「その場」観測では，中間スケールの不安定性において MHD 方程式で記述できる物理過程と粒子の運動論方程式によって記述できる物理過程が結合していることを示唆している．これらのマルチスケールの磁気圏変動である宇宙天気を真に理解するためには，全てのスケールをシームレスに扱える運動論方程式（第一原理）によるシミュレーションが本質的である．

プラズマの運動論シミュレーションには 2 つの手法がある．1 つは，プラズマ粒子であるイオンや電子などの個々の荷電粒子の運動を，ニュートンーローレンツ方程式により解き進める PIC（Particle-In-Cell）法である．格子点（Cell）上に定義された電磁場中を粒子が動きまわることから，このように呼ばれている．宇宙空間に存在する膨大な数の荷電粒子を有限の計算機資源で扱うことは不可能であるため，ある程度まとまった数の荷電粒子の集団を 1 つの“超”粒子として扱う．PIC 法はその数値解法の完成度が高く，プラズマ科学分野では広く用いられている．しかし，プラズ

†1 名古屋大学太陽地球環境研究所
Solar-Terrestrial Environment Laboratory, Nagoya University
†2 九州大学情報基盤研究開発センター
Research Institute for Information and Technology, Kyushu University

マを超粒子として扱うことにより熱雑音が大きくなること，電荷密度や電流密度などの荷電粒子の運動に起因する場の量を格子上に割り振る際に生じる高波数モードが数値誤差として蓄積すること，さらに並列化の際に負荷のバランス（各プロセス内の粒子数の均一性）を保つために特殊なデータの分割が必要になることなどの欠点がある．

一方もう 1 つの手法であるブラソフ法は，位置－速度位相空間に定義されたプラズマ粒子の分布関数の発展をブラソフ方程式により直接解き進める方法である．格子点上に定義された分布関数は熱雑音を持たず，また流体シミュレーションと同様に並列計算も容易である．しかし，ブラソフ方程式は実空間 3 次元及び速度空間 3 次元の計 6 次元を扱う方程式であり，コンピュータで解くには膨大なリソースを必要とする．このため，その手法の開発はあまり進んでいない．実際，ここ数年の HPC プロジェクトによる計算機環境の飛躍的に向上によって手法の開発が進み，実空間 2 次元及び速度空間 3 次元の 5 次元シミュレーションがようやく実用の域に達しつつある段階である．

本研究の最終的な目的は，プラズマシミュレーションとしては「次々々」世代の技術にあたる第一原理ブラソフシミュレーション手法を世界に先駆けて確立し，プラズマ科学に基づいた宇宙天気の実現に貢献することにある．そのための準備として，本研究では特に，現存する超並列計算機上のおける 5 次元ブラソフコードの性能評価及び性能チューニングを行う．

2. 計算手法の概要

2.1 基礎方程式

無衝突プラズマの振る舞いは，以下のブラソフ（無衝突ボルツマン）方程式によって記述される．

$$\frac{\partial f_s}{\partial t}+\vec{v}\frac{\partial f_s}{\partial \vec{r}}+\frac{q_s}{m_s}(\vec{E}+\vec{v}\times\vec{B})\frac{\partial f_s}{\partial \vec{v}}=0 \tag{1}$$

ここで $\vec{E}$，$\vec{B}$，$\vec{r}$ と $\vec{v}$ はそれぞれ電場，磁場，位置，速度を表す．また，$f_s(\vec{r},\vec{v},t)$ は位置-速度位相空間におけるプラズマ粒子の分布関数であり，s はイオンや電子など種類を示す．q_s と m_s はそれぞれ電荷と質量を表す．

プラズマ粒子の分布関数は，電磁場によって変形する．電磁場の時空間発展は以下のマックスウェル方程式によって記述される．

$$\nabla\times\vec{B}=\mu_0\vec{J}+\frac{1}{c^2}\frac{\partial\vec{E}}{\partial t} \tag{2.1}$$

$$\nabla\times\vec{E}=-\frac{\partial\vec{B}}{\partial t} \tag{2.2}$$

$$\nabla\cdot\vec{E}=\frac{\rho}{\varepsilon_0} \tag{2.3}$$

$$\nabla\cdot\vec{B}=0 \tag{2.4}$$

ここで $\vec{J}$ は電流密度，ρ は電荷密度，μ_0 は真空中の透磁率，ε_0 は真空中の誘電率，c は光速を示す．ブラソフ方程式(1)を速度空間で積分すると，以下の電荷保存則が得られる．

$$\frac{\partial\rho}{\partial t}+\nabla\cdot\vec{J}=0 \tag{3}$$

マックスウェル方程式(2.1)に含まれる電流密度 $\vec{J}$ はプラズマの運動によって生じ，これにより電磁場が変化する．電流密度 $\vec{J}$ はブラソフ方程式(1)の第二項にあたる実空間の流束 $\vec{v}f_s$ を速度空間で積分することによって求まり，電流密度 $\vec{J}$ が電荷保存則(3)を満足する限り，ポアソン方程式(2.3)は自動的に満たされる．

以上の方程式は，ブラソフコードにおいて解いているプラズマ粒子の運動論方程式であり，無衝突プラズマの第一原理と呼ぶ．

2.2 演算子分離法と保存型解法

ブラソフ方程式は 4 次元以上の「超次元」を扱う方程式であり，そのままの形で多次元数値積分を行うのは非常に困難であるため，演算子分離（operator splitting）法が古くから用いられてきた[1]．過去の研究では，各次元(x, y, z, v_x, v_y, v_z)それぞれを 1 次元移流方程式に分解する方法が採用されていたが，本研究では，以下のように実空間移流，速度空間移流，速度空間回転の 3 つの物理的な演算子に分離する手法を開発した[2].

$$\frac{\partial f_s}{\partial t}+\vec{v}\frac{\partial f_s}{\partial \vec{r}}=0 \tag{4.1}$$

$$\frac{\partial f_s}{\partial t}+\frac{q_s}{m_s}\vec{E}\frac{\partial f_s}{\partial \vec{v}}=0 \tag{4.2}$$

$$\frac{\partial f_s}{\partial t}+\frac{q_s}{m_s}(\vec{v}\times\vec{B})\frac{\partial f_s}{\partial \vec{v}}=0 \tag{4.3}$$

この演算子分離は，PIC 法においてニュートンーローレンツ式（荷電粒子の運動方程式）を時間 2 次精度で解く手法として広く用いられている Boris アルゴリズム[3]に基づいている．

式(4.1)及び(4.2)は多次元の移流方程式であり、$\vec{v}$ 及び $\vec{E}$ がそれぞれ $\vec{r}$ 及び $\vec{v}$ に依存しない線形方程式である．ブラソフ方程式の数値積分では，古くからセミラグランジュ法が採用されてきた．即ち，格子点上に与えられている分布関数の値を更新する際に，次の時刻にある点にくるべき分布関数の値の座標をバックトレースし，隣接する格子点より値を内挿している．2000 年代前半までは，スプライン補間や CIP（Constraint Interpolation Profile）法などの高次補間法を用いたセミラグランジュ法が用いられ，実空間 1 次元，速度空間 1 次元の，2 次元位相空間シミュレーションが主流であった．

本研究をはじめとする近年の研究では保存型セミラグランジュ法を採用し，超多次元シミュレーションに成功している．これは，保存則であるブラソフ方程式において，プラズマの数密度の保存を数値的に保証し，数値誤差を軽減しようとするものである．PIC シミュレーションでは，電荷保存則(3)を厳密に満たすことによって数値誤差の軽減を行っており，ブラソフシミュレーションでは保存型解法を用いることにより，式(3)が自動的に満たされる．そのため, 保存型解法を採用するメリットは大きい. また式(4.1)及び(4.2)の多次元の線形移流方程式を解く上で，計算精度を上げることは非常に重要であるが，2 次精度以上の手法を用いた際に現れる数値振動を抑制するためのリミッタを保存型解法に導入するのは比較的容易である．

本研究では，演算子分離による数値拡散を抑制するために，多次元の線形移流方程式に対する演算子非分離（unspliting）法を新たに開発している[2]. また本研究では，無振動性及び正値性を保証するリミッタを新たに開発し，数値振動の抑制を行っている[4][5]. ここで無振動スキームとは，ある区間において新たな極値（極大，極小）を生じず，既に存在する極値は（できるだけ）減衰させないスキームであり，ENO 法はこれに該当するが，TVD 法は極地を鈍らせるために該当しない．本研究で用いている保存型無振動スキームでは, 流束を計算する点に対して風上 3 点，風下 2 点を使用して区間プロファイルの形状を推定し，極地を検出する（図 1 参照）．数値流束は 3 次多項式[4]または 4 次多項式[5]を用いてラグランジュ補間より求め，極値に合わせたリミッタにより流束に制限を掛けている．本研究で行ったベンチマークテスト及び最近の 5 次元シミュレーションでは，5 次多項式を用いた補間をテスト的に採用しているが，この手法は未だ開発途上である．

式(4.3)は荷電粒子の速度が磁力線により運動エネルギーを保ったまま変化する回転方程式を表す．直交座標系における回転方程式は剛体回転問題と等価であり，線形移流問題と同様に，数値計算において最も基本的であるが，計算精度が重要となる問題である．本研究で採用している back-substitution 法[6]では，Boris アルゴリズム[3]に基づいて速度空間での粒子の軌道をバックトレースし，v_x, v_y, v_z 方向それぞれの演算子を分離して回転運動を解いている．剛体回転問題では，系の外側，即ち速度空間において速度が速くなればなるほど移動量（加速）は大きくなり，クーラン条件の影響を受けやすくなる点に注意が必要であり，今後，陰解法や演算子非分離法の開発が必要である．

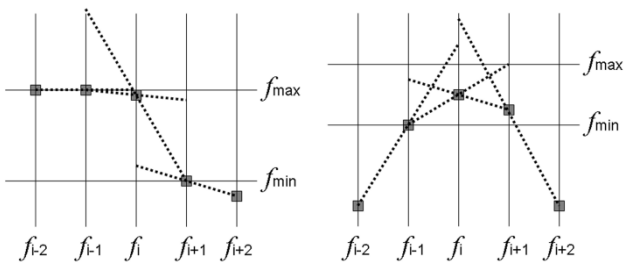

図 1　区間プロファイルにおける極大，極小の検出[4].
Figure 1　Detection of local extrema in a piece-wise profile [4].

以上のように，ブラソフ方程式の数値解法は未だ発展途上である．この大きな原因は，ブラソフコードで扱う次元が多いためであり，開発やデバッグにために大容量の共有メモリ環境が必要となるからである．

一方，マックスウェル方程式(2.1)及び(2.2)は，FDTD（Finite Difference Time Domain）法と呼ばれる電磁場解析法を用いて解く．FDTD 法では，Yee 格子[7]と呼ばれる staggered 格子を用いており，式(2.4)が自動的に満たされるように物理量が配置されている．また leap-frog アルゴリズムに基づいて電場と磁場を半タイムステップずらしており，時空間精度は 2 次である．

ブラソフシミュレーションでは非常に多くのメモリを必要とするため，並列計算が必須となる．ブラソフコードで使用する物理量は全て格子点上で与えられており，並列化においては領域分割法が有効である．図 2 は実空間 2 次元及び速度空間 3 次元を使用するブラソフコードにおける並列化の概念を示す．我々の目は 4 次元以上の空間を認識できないが, 2 次元実空間の各格子上に 3 次元速度空間（速度分布関数）が定義されていると考えると分かりやすい．本研究では図 2 のように実空間（x－y 平面）においてのみ領域分割を行い，速度空間の領域分割は行わない[8]．これは，電荷密度や電流密度などのモーメント量を計算する際に必要な速度空間の積分において，各実空間でのリダクション処理を行わないようにするためである．またオプションとして，プロセスエレメント（PE）毎に OpenMP によるスレッド並列化も許可している．

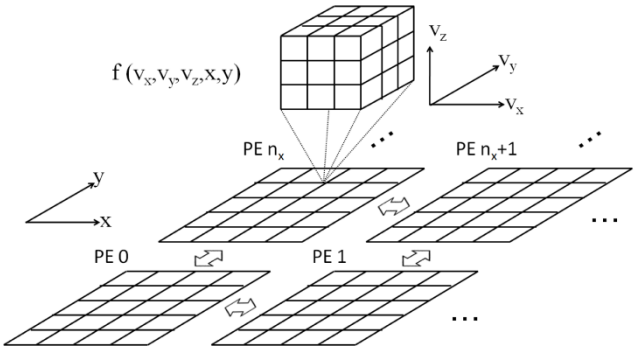

図 2　5 次元ブラソフコードにおける空間領域分割[8].
Figure 2　The domain decomposition in the configuration space for the five-dimensional Vlasov code [8].

3. 性能チューニング

図 3 に 5 次元ブラソフコードのコア演算部のプログラム例を示す．実際には図 3 のプログラムの外側に，実空間座標 x,y 及び粒子種に関する 3 重ループ(ii, jj, ss)が存在するが，ここでは省略している．基本的な演算の流れは，加速度の計算(8 行目)と加速度に基づくデータの番地の計算(9-14 行目)とデータの読み込み(16-20 行目)及び、数値フラックスの計算(22 行目)と分布関数の更新(25-29 行目)である．尚このプログラムを用いた場合，Fujitsu FX1(SPARC64 VII)では 14.%，Fujitsu CX400(Sandy Bridge)では 21%，Fujitsu FX10／京では 13%程度の実効効率であった．

図 3 のプログラムには，最内側ループ内に floor 関数の呼び出し(10 行目)とユーザ定義関数(22)行目が含まれており，コンパイラの最適化を妨げている．性能チューニングとして，まず 15 行目においてループを 2 つに分割した(図 4)．この時，ループの分割及びループ間のデータの受け渡しは 2 次元または 1 次元で行うのが適当であり，3 次元で行った場合にはデータの受け渡しに用いる配列へのアクセス負荷が大きくなって性能が極端に劣化した．このプログラムを用いた場合，Fujitsu FX1(SPARC64 VII)では 15.%，Fujitsu CX400(Sandy Bridge)では 22%，Fujitsu FX10／京では 14%程度の実効効率になり，全体的に性能が向上した．

```
1 do nn=2,nvzp1
2  uz1=vz(nn,ss)
3  do mm=1,nvyp1
4   uy1=vy(mm,ss)
5   do ll=2,nvxp1
6    ux1=vx(ll,ss)
7
8    vv = ux1*bbx+uy1*bby+uz1*bbz
9    mv = sv
10   mp0=mm-floor(vv)
11   mp2=min(max(mp0+mv+mv,1),nvyp2)
12   mp1=min(max(mp0+mv    ,1),nvyp2)
13   mm1=min(max(mp0-mv    ,1),nvyp2)
14   mm2=min(max(mp0-mv-mv,1),nvyp2)
15
16   hp2=ff(ll,mp2,nn,ii,jj,ss)
17   hp1=ff(ll,mp1,nn,ii,jj,ss)
18   hp0=ff(ll,mp0,nn,ii,jj,ss)
19   hm1=ff(ll,mm1,nn,ii,jj,ss)
20   hm2=ff(ll,mm2,nn,ii,jj,ss)
21
22  dfy(ll,mm)=flux(hp2,hp1,hp0,hm1,hm2,vv)
23  end do
24 end do
25 do mm=2,nvyp1
26   ff(2:nvxp1,mm,nn,ii,jj,ss) &
27   =ff(2:nvxp1,mm,nn,ii,jj,ss) &
28   -(dfy(2:nvxp1,mm)-dfy(2:nvxp1,mm-1))
29 end do
30 end do
```

図 3　5 次元ブラソフコードのオリジナルプログラム．

Figure 3　An original sample program of 3D Vlasov code.

```
1 do nn=2,nvzp1
2  uz1=vz(nn,ss)
3  do mm=1,nvyp1
4   uy1=vy(mm,ss)
5   do ll=2,nvxp1
6    ux1=vx(ll,ss)
7
8    vv = ux1*bbx+uy1*bby+uz1*bbz
9    mv = sv
10   mp0(ll,mm)=mm-floor(vv)
11   mp2(ll,mm)=min(max(mp0+mv+mv,1),nvyp2)
12   mp1(ll,mm)=min(max(mp0+mv    ,1),nvyp2)
13   mm1(ll,mm)=min(max(mp0-mv    ,1),nvyp2)
14   mm2(ll,mm)=min(max(mp0-mv-mv,1),nvyp2)
15   t_vv(ll,mm)=vv
16  end do
17 end do
18 do mm=1,nvyp1
19  do ll=2,nvxp1
20   hp2=ff(ll,mp2(ll,mm),nn,ii,jj,ss)
21   hp1=ff(ll,mp1(ll,mm),nn,ii,jj,ss)
22   hp0=ff(ll,mp0(ll,mm),nn,ii,jj,ss)
23   hm1=ff(ll,mm1(ll,mm),nn,ii,jj,ss)
24   hm2=ff(ll,mm2(ll,mm),nn,ii,jj,ss)
25   vv = t_vv(ll,mm)
26  dfy(ll,mm)=flux(hp2,hp1,hp0,hm1,hm2,vv)
27  end do
28 end do
29 do mm=2,nvyp1
30   ff(2:nvxp1,mm,nn,ii,jj,ss) &
31   =ff(2:nvxp1,mm,nn,ii,jj,ss) &
32   -(dfy(2:nvxp1,mm)-dfy(2:nvxp1,mm-1))
33 end do
34 end do
```

図 4　ループ分割を行った 5 次元ブラソフコードのプログラム．

Figure 4　A sample program of 3D Vlasov code with a loop decompositon.

更に，図 5 のように，図 4 の 11-14 行目の整数関数を用いた演算を倍精度実数演算に変更する．このとき，Fujitsu FX10／京では実効効率が 17%程度と大きく向上した．一方で, FX1 及び CX400 では図 3 のプログラムと同程度の実効効率となり，図 4 のチューニングの効果が無くなった．これは型変換を行うことによって演算数が増加したことが原因であるが，逆に言うと，FX10／京では整数関数の演算が極端に遅いことを意味している．

```
11 wnvyp2=nvyp2
12 wmp0=mp0(ll,mm)
13 sv=mv
14 mp2(ll,mm)=min(max(wmp0+sv+sv,1.),wnvyp2)
15 mp1(ll,mm)=min(max(wmp0+sv   ,1.),wnvyp2)
16 mm1(ll,mm)=min(max(wmp0-sv   ,1.),wnvyp2)
17 mm2(ll,mm)=min(max(wmp0-sv-sv,1.),wnvyp2)
```

図 5　整数関数を用いた演算の倍精度実数関数への型変換.
Figure 5　Conversion of integer operations to double-precision operations.

4. 性能評価

4.1 ベンチマーク条件及び計測システム

4 次元以上の超次元問題のシミュレーションには非常に多くのメモリ容量を必要とするため，コアあたりあるいはノードあたりのメモリ容量を適切に設定する必要がある．現存する計算機上で実際にシミュレーションを実行する際には，30^3—60^3 程度の速度空間を使用している．コアあたりのメモリ容量は 1GB—4GB を想定しており，速度空間の格子数と実空間の解像度を固定したまま，実空間の格子数(計算領域)を増やす，弱いスケーリングにより実際の計算を行っており，性能測定においてもこれを採用する．本研究では，コアあたりの格子数を実空間では 40×20，速度空間では 30×30×30 に設定する. これはメモリ容量で約 1GB であり, 実際の計算で使用している格子数と同程度である.

本研究で使用した超並列計算機システムは以下のとおりである. 名古屋大学及び JAXA の Fujitsu FX1 は京コンピュータのプロト機種であり，ノードあたり SPARC64VII (2.5GHz，4 コア)を 1CPU，メモリを 32GB (JAXA は 16GB) 搭載している．ノード間は DDR Infiniband (16Gbps)で接続されている．東京大学及び九州大学の Fujitsu FX10 は京コンピュータの後継機種であり，ノードあたり SPARC64IXfx (1.848GHz，16 コア)を 1CPU，メモリを 32GB 搭載している．ノード間は Tofu と呼ばれる 6 次元メッシュトーラス (160Gbps)が 4 リンクで接続されている．九州大学の Fujitsu CX400 は，ノードあたり Xeon E5-2680 (2.7GHz，8 コア)を 2CPU，メモリを 128GB 搭載している．ノード間は FDR Infiniband (54Gbps)で接続されている．最後に，理化学研究所の京コンピュータは，ノードあたり SPARC64VIIIfx (2.0GHz，8 コア)を 1CPU，メモリを 16GB 搭載している．ノード間は 6 次元メッシュトーラス(Tofu)が 10 リンクで接続されている．

4.2 ベンチマーク結果

1 コアあたり 1GB にメモリ使用量を固定したときの，5 次元ブラソフコードの弱いスケーリング性能を図 6 及び図 7 に示す．図 6 はコア数に対する実効計算速度を表し，図 7 はコア数に対する実効並列化率を表す．どのシステムにおいても，数万コアまでは性能がほぼスケールしていることが分かる．

図 7 より，Tofu ネットワーク上では数万コアまでは 90%以上のスケーラビリティを達成している．ここで，九州大学(12,288 コアまで)と東京大学(76,800 コアまで)とで性能に差があるのは，コンパイラバージョンの違いである．また，京コンピュータでは 65,536 ノード以上でスケーラビリティが大きく低下した．これは，電磁場計算における収束レベルのチェックに用いている MPI_Allreduce の通信時間が，ノード数(コア数ではなく)の増加によって大きく増加するためである. 事実, 京コンピュータの 65,536 コア(8192 ノード)よりも FX10 の 76,800 コア(4800 ノード)のほうが高い実効性能が得られた.

一方で CX400 では，4096 コアまでは 19%以上の実効性能であったが，16,384 コア(1024 ノード)以上から性能が劣化した．これは，九州大学の CX400 が 256 ノードごとにフルバイセクションバンド幅で接続された構成になっており，全ノードがフルバイセクションバンド幅で接続されていないためである.

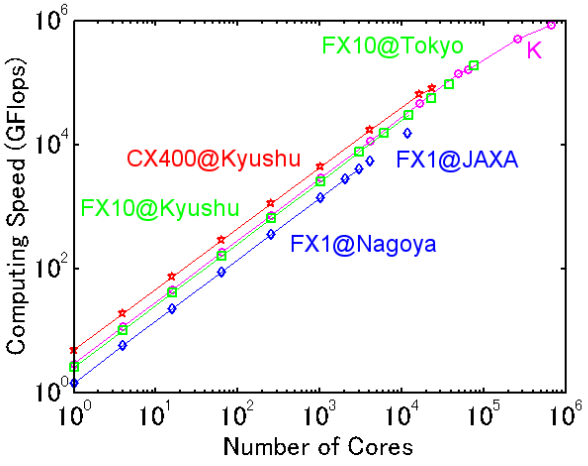

図 6　1GB/core 使用時の 5 次元ブラソフコードの弱いスケーリング性能．コア数に対する実効速度.
Figure 6　Weak-scaling performance of the 5-dimensional Vlasov code with 1GB/core. Computational speed as a function of the number of cores.

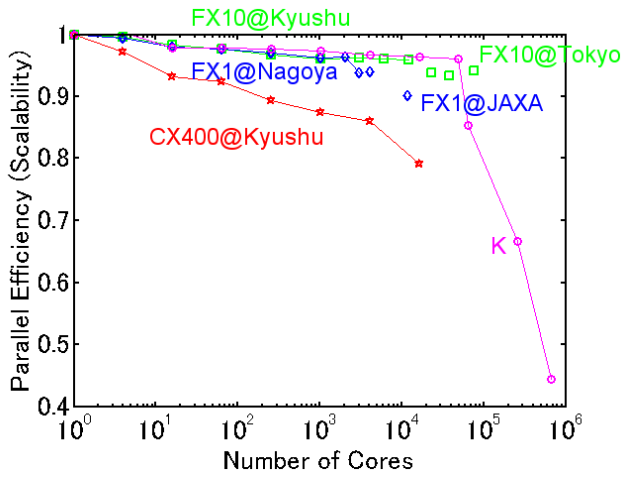

図 7　1GB/core 使用時の 5 次元ブラソフコードの弱いスケーリング性能．コア数に対する実効並列化率．

Figure 7　Weak-scaling performance of the 5-dimensional Vlasov code with 1GB/core. Scalability as a function of the number of cores.

5. おわりに

ブラソフコードは，宇宙空間に広く存在する無衝突プラズマの第一原理シミュレーション手法である．プラズマは位置－速度位相空間における分布関数として定義され，超多次元関数として与えられる．ブラソフシミュレーションは計算負荷が非常に高く，その手法の開発やデバッグが困難であるため，計算手法は未だ発展途上にある．本研究では，新たに開発した 2 次元実空間及び 3 次元速度空間を扱う 5 次元ブラソフコードについて，国内の代表的な超並列計算機における性能評価及び性能チューニングを行った．チューニングしたコードは，6000 ノードまでの並列計算においても実効効率 15%以上，実効並列化率 90%以上の高い性能を得たが，8000 ノード以上において MPI_Allreduce の通信性能の劣化が見られた．

謝辞　本研究は，科学研究費補助金・挑戦的萌芽研究 No.25610144 によりサポートを受けている．ベンチマークテストに使用したスーパーコンピュータシステムの計算リソースは，名古屋大学太陽地球環境研究所計算機利用共同研究，名古屋大学 HPC 連携研究プロジェクト，東京大学大規模 HPC チャレンジ，九州大学先端的計算科学研究プロジェクト，学際大規模共同利用・共同研究(jh130005) 及び，HPCI システム利用研究(hp120092)により提供された．また性能チューニングに際し，サイエンティフィック・システムズ研究会マルチコア性能 WG 及び富士通に協力を頂いた．

参考文献

1. Cheng, C. Z., Knorr, G. : The integration of the Vlasov equation in configuration space, *J. Comput. Phys.*, Vol.22, No.3, 330—351 (1976).
2. Umeda, T., Togano, K., Ogino, T.: Two-dimensional full-electromagnetic Vlasov code with conservative scheme and its application to magnetic reconnection, *Comput. Phys. Commun*., Vol.180, No.3, 365—374 (2009).
3. Boris, J. P.: Relativistic plasma simulation-optimization of a hybrid code, *Proc. Fourth Conf. Num. Sim. Plasmas*, ed. by J. P. Boris and R. A. Shanny, pp.3—67, Naval Research Laboratory, Washington D. C. (Nov. 1970).
4. Umeda, T.: A conservative and non-oscillatory scheme for Vlasov code simulations, *Earth Planets Space*, Vol.60, No.7, 773—779 (2008).
5. Umeda, T., Nariyuki, Y., Kariya, D.: A non-oscillatory and conservative semi-Lagrangian scheme with fourth-degree polynomial interpolation for solving the Vlasov equation, *Comput. Phys. Commun.*, Vol.183, No.5, 1094—1100 (2012).
6. Schmitz, H., R. Grauer, R.: Comparison of time splitting and backsubstitution methods for integrating Vlasov's equation with magnetic fields, *Comput.Phys. Commun*., Vol.175, No.2, 86—92 (2006).
7. Yee, K. S., Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antenn. Propagat.*, Nol.AP-14, No.3, 302—307 (1966).
8. Umeda, T., Fukazawa, K., Nariyuki, Y., Ogino, T.: A scalable full electromagnetic Vlasov solver for cross-scale coupling in space plasma, *IEEE Trans. Plasma Sci.*, Vol.40, No.5, 1421—1428 (2012).